

Multi-Disciplinary, Multi-Fidelity Discrete Data Transfer Using Degenerate Geometry Forms

Erik D. Olson*

NASA Langley Research Center, Hampton, VA 23681

In a typical multi-fidelity design process, different levels of geometric abstraction are used for different analysis methods, and transitioning from one phase of design to the next often requires a complete re-creation of the geometry. To maintain consistency between lower-order and higher-order analysis results, Vehicle Sketch Pad (OpenVSP) recently introduced the ability to generate and export several degenerate forms of the geometry, representing the type of abstraction required to perform low- to medium-order analysis for a range of aeronautical disciplines. In this research, the functionality of these degenerate models was extended, so that in addition to serving as repositories for the geometric information that is required as input to an analysis, the degenerate models can also store the results of that analysis mapped back onto the geometric nodes. At the same time, the results are also mapped indirectly onto the nodes of lower-order degenerate models using a process called aggregation, and onto higher-order models using a process called disaggregation. The mapped analysis results are available for use by any subsequent analysis in an integrated design and analysis process. A simple multi-fidelity analysis process for a single-aisle subsonic transport aircraft is used as an example case to demonstrate the value of the approach.

Nomenclature

C_{D_0}	profile-drag coefficient
C_D	drag coefficient
c_d	sectional drag coefficient
$c_{d,f}$	drag coefficient due to skin friction
c_{d_0}	sectional profile-drag coefficient
c_f	skin-friction coefficient
\bar{c}_f	average skin-friction coefficient
C_L	lift coefficient
c_l	sectional lift coefficient
$c_{l_{\max}}$	sectional maximum lift coefficient
c_{l_0}	sectional lift coefficient at zero angle of attack
c_{l_α}	sectional lift-curve slope
C_M	pitching-moment coefficient
c_m	sectional pitching-moment coefficient
c_{m_0}	sectional pitching-moment coefficient at zero angle of attack
c_p	pressure coefficient
h	altitude
M_∞	free-stream Mach number
S	wing reference area
u, w	parametric coordinates
V_{eas}	equivalent airspeed
x, y, z	Cartesian coordinates
α	angle of attack

*Aerospace Engineer, Aeronautics Systems Analysis Branch, Senior Member AIAA.

ϕ, θ, ψ	Euler rotation angles for a section
Subscripts	
$2D$	values calculated by two-dimensional analysis
$3D$	values calculated by three-dimensional analysis

I. Introduction

DURING a traditional aircraft design process, early conceptual design studies are performed using *lower-order* (a.k.a. lower-fidelity) analysis methods with a limited number of degrees of freedom, transitioning to *higher-order* analysis as the design becomes more refined. The lower-order analysis methods are performed on simplified representations of the geometry, while higher-order methods use geometry that is much more representative of the as-built aircraft.¹ Because of this disparity in geometric forms, transitioning from one phase of design to the next requires a complete re-creation of the geometry, and the previous analysis results must be discarded along with the lower-order geometric representation. These discontinuities in the process are even more of a problem in newer aircraft design philosophies, which use *multi-fidelity* approaches that mix higher-order analysis into the conceptual design phase to supplement the results of the lower-order methods. To maintain consistency between the lower-order and higher-order analysis results, it is crucial to implement a *multi-fidelity geometry* approach that maintains a direct link between the various geometrical representations, regardless of the type of analysis method used.²

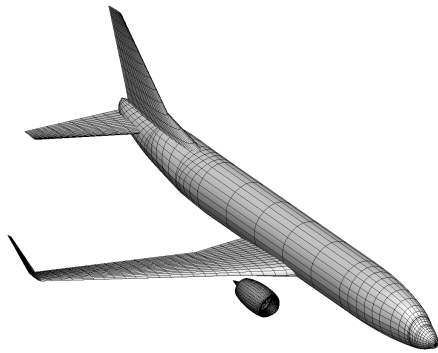
One tool that has embraced this multi-fidelity geometry approach is Vehicle Sketch Pad (OpenVSP)³ which is an open-source parametric geometry modeler that allows rapid geometry creation using high-level design parameters. OpenVSP has been used extensively for conceptual design studies of aircraft, including studies using higher-order analysis.^{4–6} To facilitate multi-fidelity analysis, OpenVSP has the ability to generate and export several *degenerate* forms of the geometry.⁷ The degenerate models represent the type of abstraction required to perform low- to medium-order analysis for a range of aeronautical disciplines. Each node in the degenerate models represents a discrete point in the geometric representation; in addition, the node retains information about the aggregate properties of the full geometry. By creating these degenerate models internally, OpenVSP maintains a direct link between the “master”, or analytical, geometry and the degenerate models, so that any changes to the design are always reflected in all of the degenerate models and in the analysis methods that employ them.

Since the nodes of the degenerate models are consistent with the geometric abstraction of the analysis method being used, this means they could also serve as natural repositories for the resulting analysis data. In this research, the functionality of OpenVSP’s degenerate geometric models is extended so that in addition to providing geometric inputs to an analysis, the degenerate models are used to store the results of that analysis, mapped directly back onto the geometric nodes from which the analysis was created. These results are then available to subsequent analyses in other disciplines, always maintaining the link between the geometry and the associated analysis results. The results are also simultaneously mapped onto the nodes of all the other degenerate models; the data are then available to both higher- and lower-order analyses in whatever level of abstraction they require, regardless of the degenerate model on which the original analysis was based. This internal data storage capability, combined with the automated data mapping algorithms, can greatly facilitate the creation of multi-disciplinary, multi-fidelity analysis, design and optimization processes with OpenVSP as the geometry modeler.

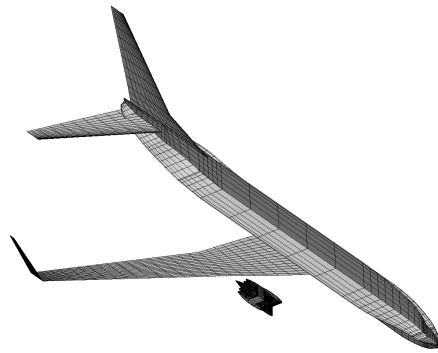
II. Degenerate Geometric Forms

The degenerate models exported from OpenVSP are at four different *orders*, or levels of abstraction. These levels are, in decreasing order: Surface, Plate, Stick, and Point⁷ (Fig. 1).

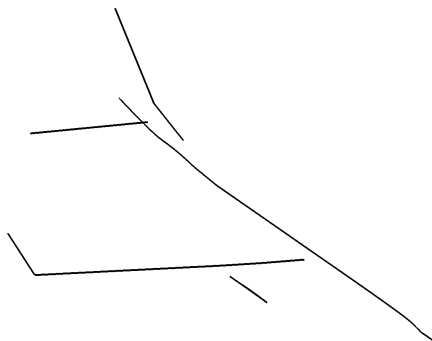
The *Degenerate Surface* (Fig. 1(a)) represents the discrete surface nodes of a component mapped against parametric u and w coordinates, and is a suitable level of abstraction for three-dimensional analysis methods, such as an aerodynamic panel method or acoustic scattering code. The nodes of the Surface model are a set of non-intersected, structured meshes for the individual components of the model (such as fuselage, wing, and tails). In addition to their three-dimensional x, y, z coordinates, the nodes maintain information about the areas and surface normals of the quadrilateral faces that are bounded by them.



(a) Surface model



(b) Plate model



(c) Stick model



(d) Point model

Figure 1. OpenVSP degenerate geometric models for a tube-and-wing aircraft configuration

The *Degenerate Plate* (Fig. 1(b)) replaces the three-dimensional surface of each component with a zero-thickness mean camber surface mapped against u and w . The Plate model is suitable for analysis methods that assume that the surface is very thin, such as an aerodynamic vortex-lattice code, an equivalent-plate structural analysis, or a planar acoustic shielding analysis. Lifting surfaces are represented by a single Plate, while bodies are represented by a pair of Plates representing both a horizontal and vertical slicing of the component in a cruciform layout. The nodes of the Plate model maintain information about the x, y, z coordinates, plus the thickness-to-chord ratio and camber at each node.

The *Degenerate Stick* (Fig. 1(c)) represents each component as a single reference line containing information about the spanwise (for lifting surfaces) or lengthwise (for bodies) properties of the component, mapped against the u parameter. The Stick model is suitable for aerodynamic analysis using a lifting-line code, or for structural analysis using an equivalent-beam method. In addition to the x, y, z coordinates, the Stick model maintains information about the chord, incidence, maximum thickness, maximum camber, center of mass, sectional area, and mass moments of inertia at each node along the spanwise axis.

Finally, the *Degenerate Point* (Fig. 1(d)) represents each component as a single point containing information about the component as a whole. The Point model is suitable for aerodynamic analysis using empirical or semi-empirical skin-friction analysis. The Point maintains information about the surface area, volume, center of mass, and mass moments of inertia of each component.

III. Discrete Data Mapping

The previous section described the aggregate geometric information stored with each degenerate form of a given component. This research extends these data-storage capabilities so that each node is also able to store any analysis results that are associated with it, plus any data that can be derived from analysis results on the higher- or lower-order degenerate models. Whenever an analysis is performed using geometry from one of the degenerate models, the results are mapped back onto the geometry in three distinct steps:

1. The analysis results are mapped onto discrete nodes of the same order as the analysis—e.g., vortex-lattice results are mapped onto the Plate model, and lifting-line results are mapped onto the Stick model.
2. The analysis results are mapped onto the lower-order degenerate models—e.g., results from the Plate model are mapped onto the Stick model and through to the Point model.
3. The analysis results are mapped onto the higher-order degenerate models—e.g., results from the Plate model are used to construct values on the nodes of the Surface model.

The processes by which these mappings take place are described in the following sections.

A. Mapping Analysis Results onto the Degenerate Model

Each analysis method stores its results onto the nodes of the corresponding degenerate type. The analysis nodes and the geometric nodes are of the same order—e.g., they are both Plate or Stick representations—but they do not necessarily coincide since the analysis method will often re-panel the geometry to improve accuracy or convergence. Consequently, a mapping process is used that interpolates the analysis results for the values at the geometric nodes, through linear interpolation, spline fitting, or some other method.

B. Mapping to Lower-Order Degenerate Forms (Aggregation)

Mapping of data from higher- to lower-order models is accomplished through *aggregation* (Fig. 2), which is similar to the processes through which the degenerate models were originally created. Three types of aggregation are typically used, depending on the type of analysis and the degenerate models being used:

- Integration: values on the lower-order model represent the integration of a distribution of values on the higher-order model. For example, the section lift and pitching-moment coefficients on each node of the Stick model can be derived by chordwise integration of the differential pressure coefficients on the corresponding section of the Plate model, while the section drag coefficients can be derived by integrating the skin-friction coefficients.

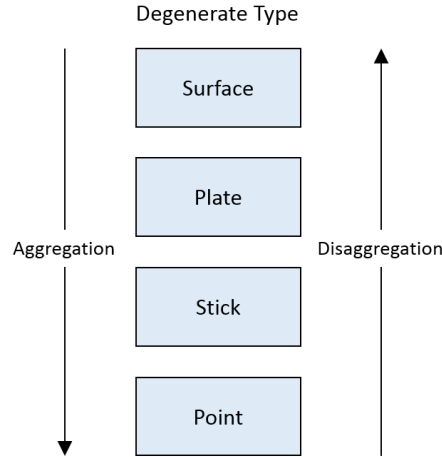


Figure 2. Aggregation and disaggregation operations

- Differencing: aggregated values represent the difference between two values. Differencing is a common aggregation process for mapping Surface values to Plate values. For example, the differential pressure coefficients on the Plate model can be derived from the difference between the pressure coefficients on the upper-surface and lower-surface nodes on the Surface model.
- Averaging: aggregated values represent the average of two or more values. For example, the skin-friction coefficients on the Plate model can be derived from the average value of the upper- and lower-surface values.

Examples of these types of aggregation will be seen in the example problem of Section V.

C. Mapping to Higher-Order Degenerate Forms (Disaggregation)

Mapping of data from lower- to higher-order models is accomplished through a *disaggregation* process (Fig. 2). In general, the disaggregation process will produce an approximation to the higher-order data since the lower-order analysis is carried out on an abstraction of the higher-order geometry. Whereas there is typically only a single type of process through which the aggregation logically is carried out between two degenerate models, in the reverse direction there can be several different disaggregation options from which to choose:

- Shape function: the distribution of values on the higher-order model may be approximated by assuming a given shape function, using knowledge about the expected distribution. For example, sectional aerodynamic coefficients (lift and moment) could be disaggregated to chordwise pressure differential at each Plate node by assuming a triangular chordwise distribution, resulting in a distribution which approximates the expected distribution that would result from an analysis of the Plate model.
- Scaling: the distribution may be approximated by scaling previously-existing higher-order analysis results to match the new lower-order results. For example, pressure coefficients for the Surface nodes could be scaled in order to produce a chordwise distribution that corresponds to the sectional lift coefficient from the corresponding node of the Stick model.
- Interpolation: the distribution may be approximated by interpolating previously-existing higher-order analysis results that are tabulated against a lower-order variable. For example, if a series of previously-computed spanwise lift distributions for the Stick model are tabulated against the overall lift coefficient, then a new distribution may be derived by interpolating against an updated lift coefficient from the Point model.
- Response-surface equation: similar to interpolation, the distribution may be approximated using regression of previously-computed distributions against one or more variables.

Regardless of the method used, it is essential that it be formulated so that it is *reversible*; that is, it should be possible to perform the reverse aggregation and arrive at the original values. For example, when lift and pitching-moment coefficients are used to derive a chordwise distribution of pressure coefficients through a reversible disaggregation process, then the resulting distribution could subsequently be integrated to produce the same lift and pitching-moment coefficients from which it was originally derived.

IV. Implementation

The mapping process described in the previous section has been implemented as a Java™ class, named DegenGeom, with an application program interface (API) that makes the various mapping methods available for use by the desired analysis methods. A DegenGeom object (i.e., an instance of the DegenGeom class) is instantiated by reading in and parsing a Degenerate Geometry file that has been exported by OpenVSP. The object can be made up of multiple components (such as wing, fuselage, etc.) and the Surface, Plate, Stick and Point models are maintained as subsets of each of these components. The different degenerate models for each component are interlinked, so that mapping analysis results onto one degenerate model can simultaneously perform aggregation of the data onto lower-order models and disaggregation onto higher-order models. The specific forms for the aggregation and disaggregation processes are defined for each quantity of interest using standard methods such as those described in the previous section, or through a custom algorithm.

To take advantage of the data mapping processes in the DegenGeom class, analysis methods are integrated using the ModelCenter® framework.⁸ The individual analyses are included in the framework using wrappers, which accept input variables from the framework, pass them to the analysis method for execution, and then provide the analysis results as output variables. These wrappers interface with the API for the DegenGeom class in order to access previous analysis results stored with the geometry, and to store the new analysis results and map them onto the degenerate models. Since the DegenGeom object stores the results of the analysis, it is normally the only output that a wrapper needs to produce. The process through which the wrapper accesses the API is illustrated in Fig. 3 for the case of an analysis that is based on the Stick model.

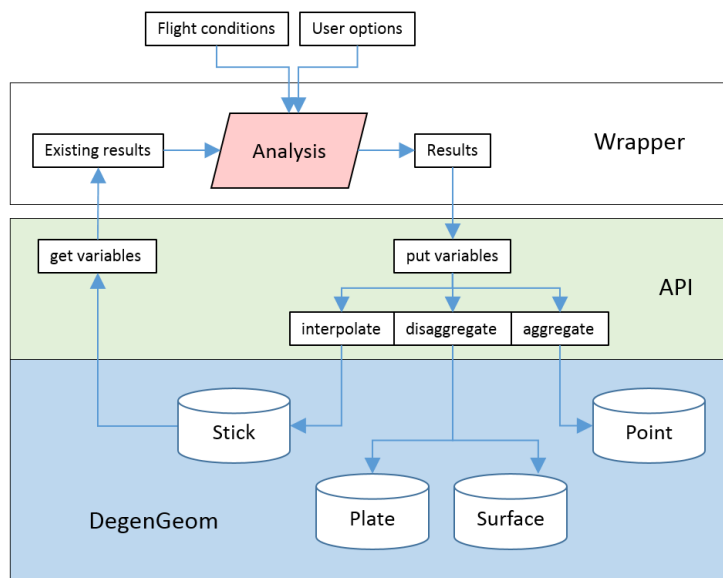


Figure 3. Process through which a wrapper accesses the API for DegenGeom, for an analysis method based on the Stick model.

The DegenGeom class is serializable, so it can be passed as output from one wrapper and reconstituted as an object in another wrapper. By accepting a DegenGeom object as input, a wrapper also has access to the results of any analysis previously performed on the same geometry. In this manner, the object is passed from one wrapper to the next, storing analysis results at each step and providing them to subsequent analysis methods where needed. Model creation is greatly simplified since almost all analysis outputs are

encapsulated in the single DegenGeom object, rather than a complex series of variables and arrays that must be linked to subsequent wrappers independently. All of the intermediate and final analysis results are available in the DegenGeom object at the end of the process and can be post-processed and plotted at that point.

V. Subsonic Transport Example

As an example case to demonstrate the utility of the mapping process, consider a single-aisle subsonic transport aircraft. The OpenVSP model is shown in Fig. 4, and the corresponding degenerate geometry models for this aircraft were shown previously in Fig. 1. Each of the separate lifting-surface components (wing, horizontal tail and vertical tail) possesses a Point, Stick, Plate and Surface degenerate representation, while each of the body components (fuselage, nacelle) possesses a Point and Surface representation and a pair of Stick and Plate representations corresponding to horizontal and vertical slices of the geometry.

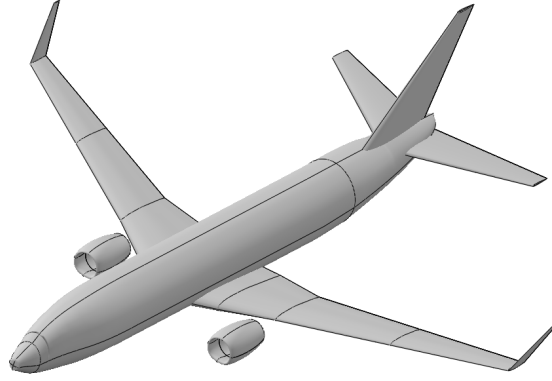


Figure 4. Single-aisle transport OpenVSP model

The methodology described in the previous section was used to build the elements of a simple multi-fidelity, multi-disciplinary aero-structural analysis of the single-aisle transport aircraft in ModelCenter[®] (Fig. 5). Each analysis method accesses the API routines of the DegenGeom class to map its analysis results onto the degenerate model, as well as simultaneously performing aggregation and disaggregation operations to map the results onto the other degenerate models, as appropriate. Each subsequent analysis method has access to the results from the previous analysis.

Analysis of the profile drag of the fuselage is conducted using the Virginia Tech FRICTION code,⁹ which applies a form factor correction to empirical skin-friction drag of a flat plate. The wetted area, reference length, and fineness ratio of the fuselage are obtained from the values that are calculated by OpenVSP and exported in the Point model. The profile-drag coefficient (C_{D_0}) computed by FRICTION is stored in the fuselage's Point model for each flight condition analyzed (Fig. 6). Table 1 shows the results of the analysis for a typical climb condition.

Table 1. FRICTION fuselage analysis results, $V_{\text{eas}} = 250 \text{ kt}$, $h = 10,000 \text{ ft}$

Quantity	Value	Units
Wetted area	4227	ft ²
Reference length	124.9	ft
Fineness ratio	9.615	
Form factor	1.058	
Reynolds number	305×10^6	
$C_{D_0} S$	7.833	ft ²
C_{D_0}	0.00603	

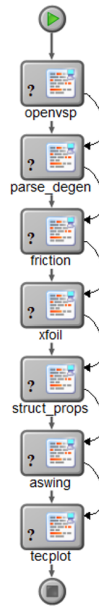


Figure 5. ModelCenter[®] model for aero-structural analysis

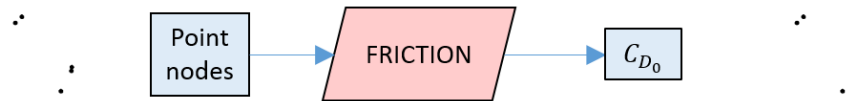


Figure 6. Process for mapping FRICITION analysis results onto the degenerate Point model.

Next, the two-dimensional wing and tail section aerodynamic polars are calculated using XFOIL, which is a panel code with tightly-coupled integral boundary-layer analysis.¹⁰ The airfoil coordinates are normalized from the Surface model for each lifting surface, and can use the actual cross sections or interpolate between them. Figure 7 gives sample results from the XFOIL analysis for the six spanwise locations along the wing where the airfoils are defined; additional results for the horizontal and vertical tails are not shown. The calculated airfoil surface properties (pressure coefficient and skin-friction coefficient as a function of angle of attack) are mapped onto the nodes of the Surface model.

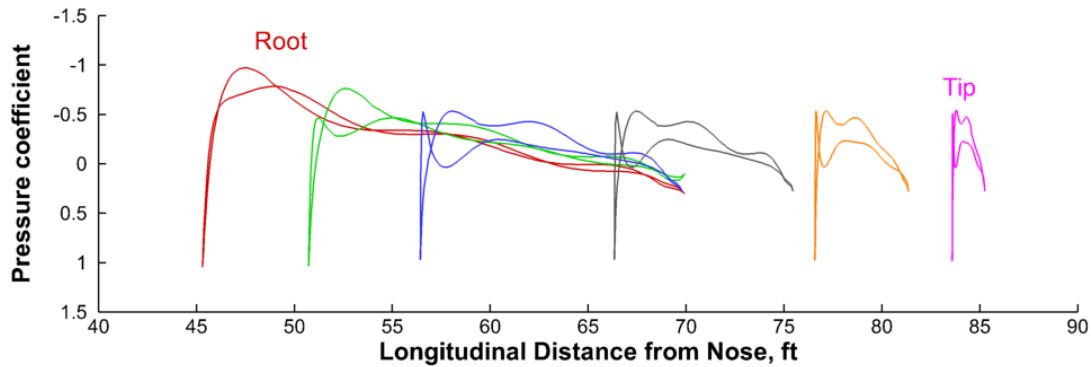


Figure 7. Pressure distributions calculated by XFOIL for cross sections of the Surface model at $\alpha = 0$, $M_\infty = 0.3$, $h = 10,000$ ft

Figure 8 illustrates the progression of steps for mapping the pressure and skin-friction coefficient values to the Surface nodes and then further aggregating and mapping them onto the lower-order models. The spanwise and chordwise variations of differential pressure coefficient are determined at each angle of attack by simply subtracting the lower-surface pressure coefficient from the upper-surface value, and then mapping the differential onto the Plate model. In addition, the surface pressure and skin-friction distributions may be aggregated into sectional lift, drag, and pitching-moment coefficients (as a function of angle of attack) through chordwise integration, and then mapped onto the Stick model. In practice, this integration is already performed by XFOIL so the coefficients calculated by the program are used instead. Figure 9 shows the calculated two-dimensional lift curve and drag polar for each of the wing sections. The spanwise variations of the two-dimensional sectional aerodynamic coefficients (c_{l_0} , c_{l_α} , c_{d_0} , and c_{m_0}) are also determined from the aerodynamic polars and mapped onto the Stick model. Since the analysis is two-dimensional, further aggregation onto the Point model is not meaningful.

As a final analysis step, an aero-structural analysis of the complete aircraft is performed using ASWING, which combines a lifting-line aerodynamic analysis with an equivalent-beam structural analysis to calculate the spanwise lift, drag, and pitching-moment distributions and the deflections and rotations of the nodes of the equivalent beam.¹¹ The aerodynamic analysis draws on the sectional coefficients that were previously calculated in XFOIL and stored with the Stick model, plus the fuselage profile-drag coefficient that was calculated in FRICTION and stored with the Point model. The structural analysis draws on the spanwise variation of the moments of inertia that is inherent to the Stick model exported from OpenVSP, combined with isotropic values of the Young's modulus and shear modulus, to derive the stiffness matrices for the Stick nodes.

Figure 10 shows the spanwise distribution of lift coefficient from the ASWING analysis for a 2.5-g pull-up maneuver at the climb flight condition. The wrapper loads the analysis results back into the DegenGeom class via the Stick model (Fig.11). The aerodynamic coefficients are the result of a three-dimensional analysis of the finite wing, distinct from the two-dimensional coefficients from the previous XFOIL analysis, and are thus bookkept separately. To map the results onto the higher-order Plate and Surface models, the lift coefficient at the Stick node is used to interpolate the previous two-dimensional analysis results from XFOIL. The mapping process interpolates for the differential pressure coefficient at each Plate node and the surface pressure coefficient at each Surface node for each case analyzed by ASWING. The drag coefficient distribution is also used to interpolate for the skin-friction coefficients on the Plate and Surface models.

Figure 12 shows the corresponding deflections of the stick model for the same 2.5-g pull-up case. The deflections and rotations of the Stick nodes from each loading case are saved as increments to the jig shape

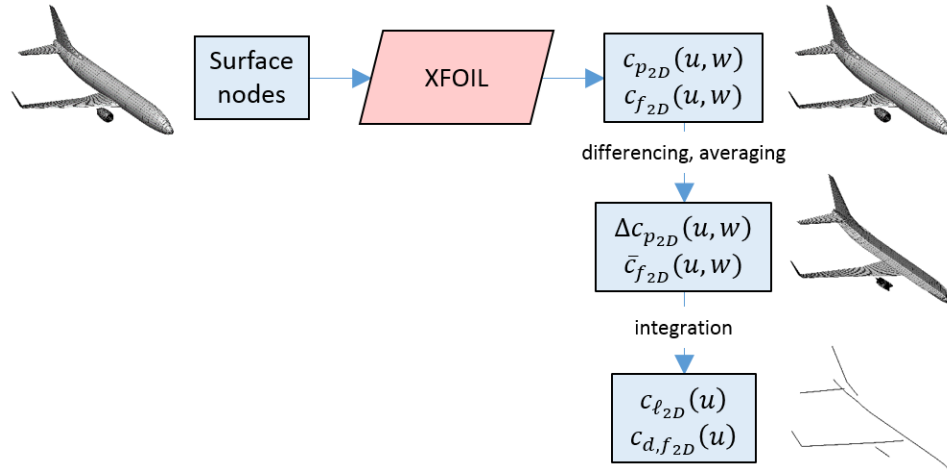
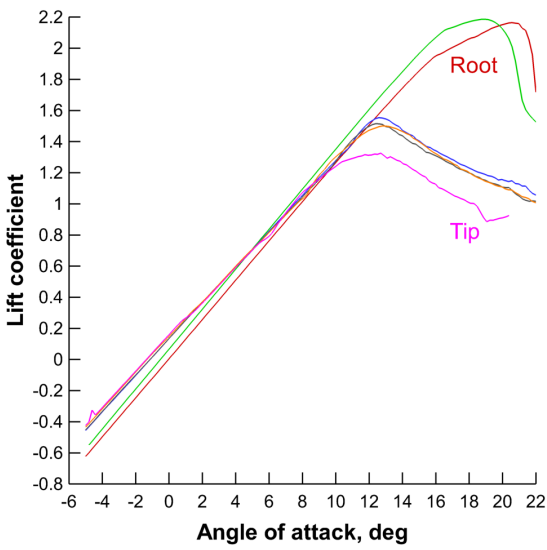
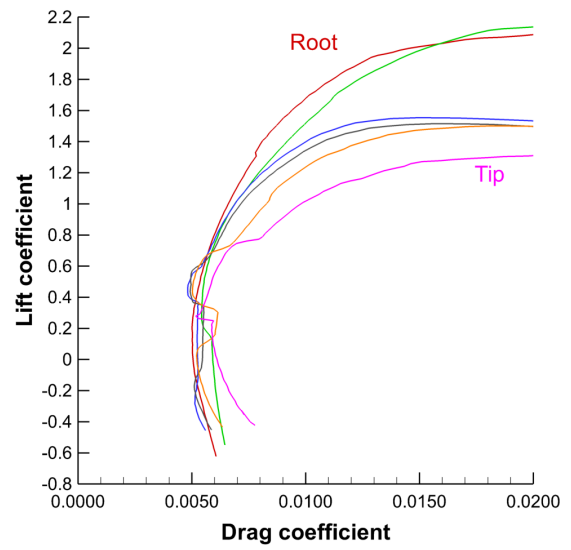


Figure 8. Process for mapping XFOIL pressure coefficients onto degenerate models.



(a) Lift curves



(b) Drag polars

Figure 9. Lift curves and drag polars calculated by XFOIL for cross sections of the Surface model at $M_\infty = 0.3$, $h = 10,000$ ft

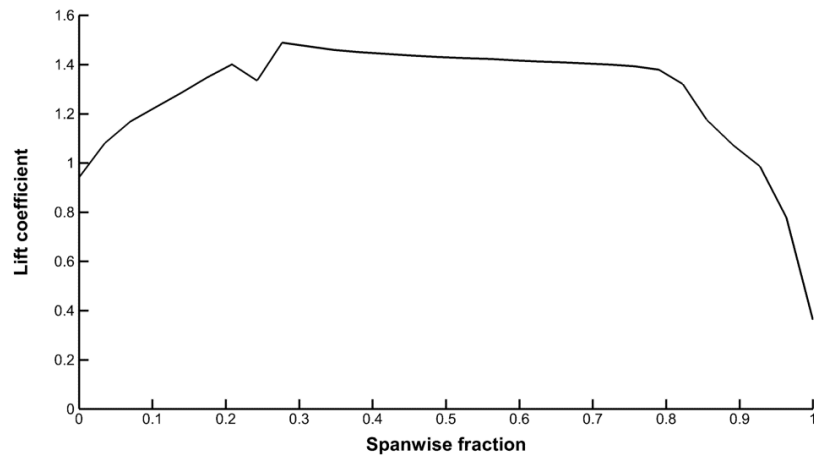


Figure 10. Spanwise lift coefficient distribution calculated by ASWING for a quasi-steady 2.5-g pull-up maneuver at $V_{\text{eas}} = 250$ kt, $h = 10,000$ ft

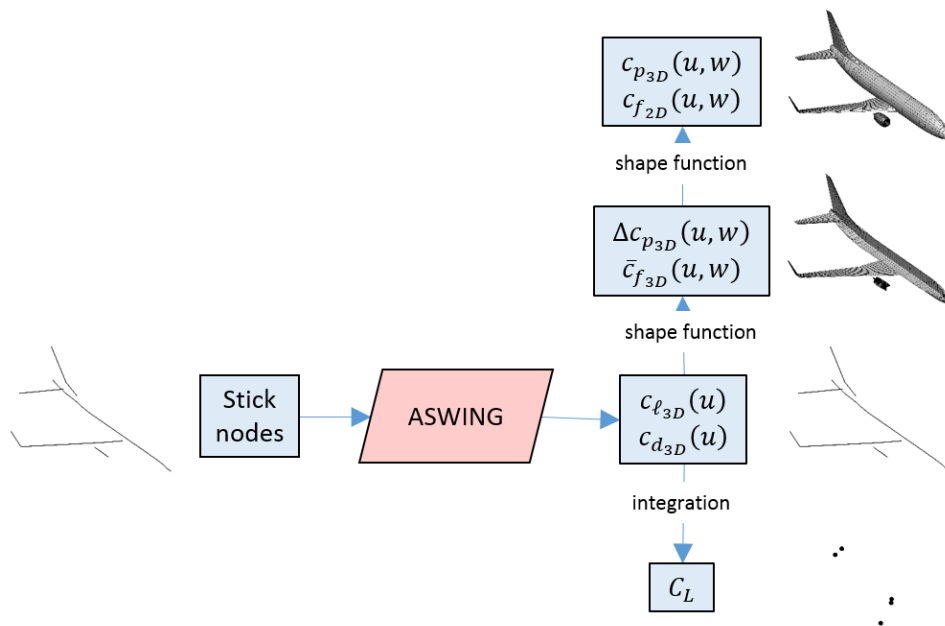


Figure 11. Process for mapping ASWING spanwise aerodynamic coefficients onto degenerate models.

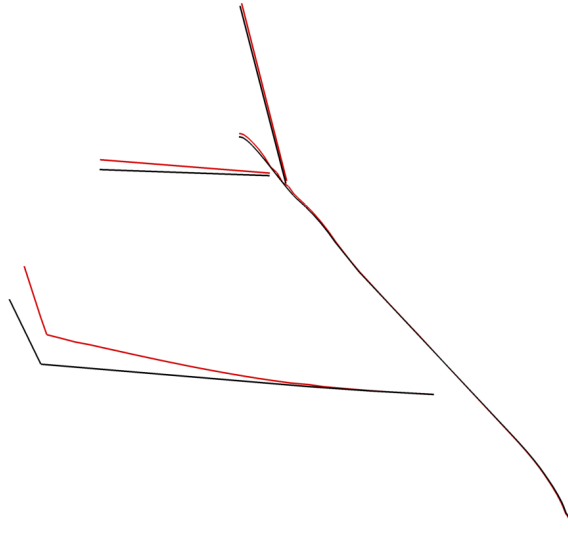


Figure 12. Stick model deflections calculated by ASWING for a quasi-steady 2.5-g pull-up maneuver at $V_{\text{eas}} = 250 \text{ kt}$, $h = 10,000 \text{ ft}$

at each node (Fig.13). The deflections and rotations are also mapped onto both the Plate and Stick models by assuming a rigid-body rotation of each cross section about the neutral axis.

Though simple, the analysis process is truly multi-fidelity: the surface pressure coefficients were calculated by a two-dimensional analysis method on the Surface model, the full wing and tail aerodynamics were calculated by a lifting-line method based on the Stick model, and the full-configuration elastic deflections were calculated by an equivalent-beam method based on the Stick model. As a result of the disaggregation process, however, both the aerodynamic and structural analysis results are automatically mapped onto all of the degenerate models—Point, Stick, Plate and Surface. It is possible, then, to plot both the surface pressure coefficients and deflections of the Surface model resulting from the 2.5-g pull-up maneuver (Fig. 14), even with pressure coefficients calculated by a two-dimensional analysis at a different flight condition, and with deflections calculated by an equivalent-beam model.

The preceding discussion gives only a sample of the results from the different analysis methods included in the process. Table 2 gives a full list of the quantities calculated by each analysis method and the corresponding degenerate model onto which they are stored. As additional analysis methods are brought into the existing process, additional quantities would be stored within the DegenGeom object, but the previous analysis results would continue to be passed forward.

Table 2. Quantities mapped onto degenerate models by each analysis method

Analysis	Point	Stick (u)	Plate (u, w)	Surface (u, w)
FRICTION	fuselage C_{D_0}	—	—	—
XFOIL	—	$c_{l_{2D}}(\alpha)$, $c_{d_{2D}}(\alpha)$, $c_{m_{2D}}(\alpha)$, c_{l_0} , c_{l_α} , $c_{l_{\max}}$, c_{d_0} , c_{m_0}	$\Delta c_{p_{2D}}(\alpha)$	$c_{p_{2D}}(\alpha)$, $c_{f_{2D}}(\alpha)$
ASWING	C_L , C_D , C_M , Δxyz (per case)	c_l , c_d , c_m , Δxyz , $\Delta \theta \phi \psi$ (per case)	Δc_p , Δxyz (per case)	c_p , c_f , Δxyz (per case)

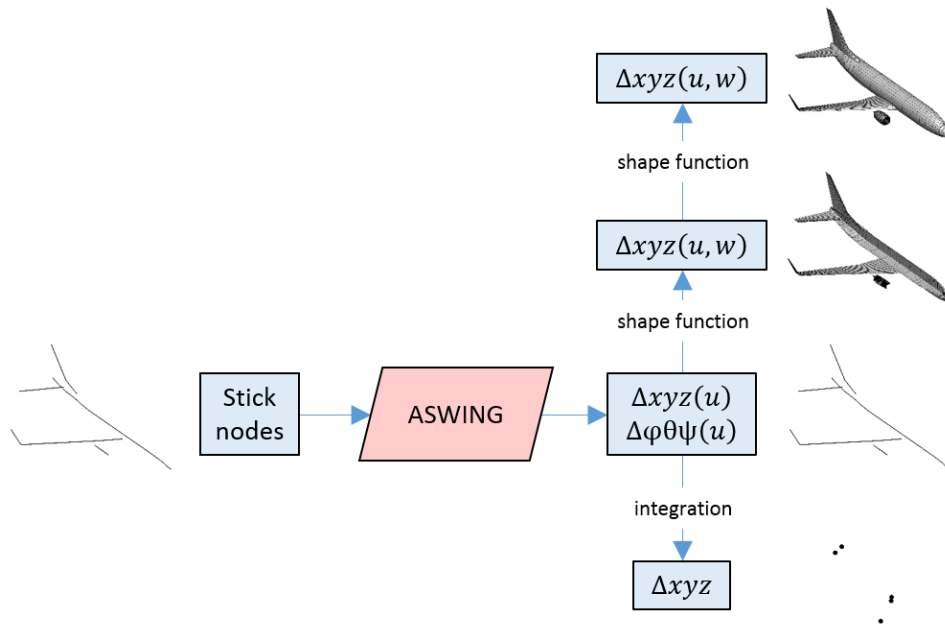


Figure 13. Process for mapping ASWING nodal deflections and rotations onto degenerate models.

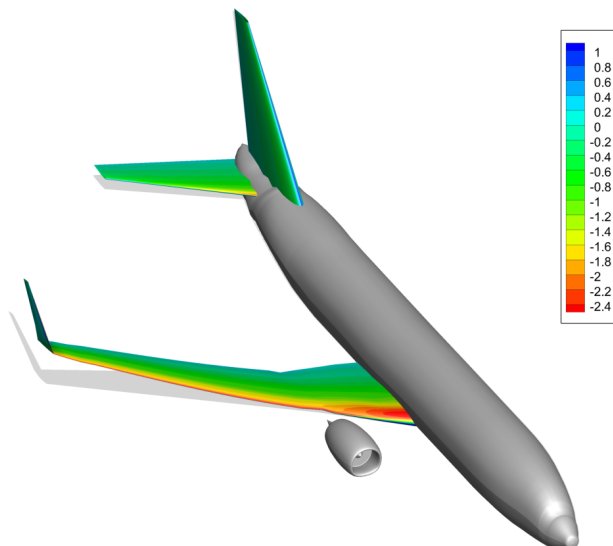


Figure 14. Deflections and pressure coefficient contours mapped onto the Surface model for a quasi-steady 2.5-g pull-up maneuver at $V_{\text{EAS}} = 250$ kt, $h = 10,000$ ft

VI. Conclusion

In this research, the functionality of OpenVSP's degenerate geometric models was extended so that in addition to serving as repositories for geometric information that can be used as inputs to an analysis, the degenerate models can also store the results of that analysis mapped back onto the geometric nodes. The results are mapped onto the same degenerate model that was used to create the analysis, but they can also simultaneously be mapped onto other lower-order models using an aggregation process, and onto higher-order models using disaggregation. A simple multi-fidelity analysis process for a single-aisle subsonic transport is used as an example case to illustrate the value of the approach.

The mapping processes were implemented as a Java™ class named DegenGeom with an API that can be accessed by wrappers in an engineering modeling framework such as ModelCenter®. A DegenGeom object can be passed from one wrapper to another within an integrated process, carrying with it the results of previous analyses mapped onto multiple degenerate forms of the geometry. Model creation is greatly simplified, since the single DegenGeom object holds all of the analysis results, rather than requiring a complex set of output variables and arrays. To date the aggregation and disaggregation processes have been formulated in a mostly ad-hoc manner depending on the specific analysis method being used, but it should be possible to further automate these processes by defining a set of universal mapping algorithms that can automatically enforce consistency and reversibility.

One shortcoming of the mapping process for the Surface model is that the components (fuselage, wing, etc.) are maintained as separate, non-intersected surfaces. The capabilities developed here could be extended significantly by also applying them to the intersected, unstructured surface mesh that can be exported by OpenVSP. The points of the intersected mesh originate from the same parametric surface as the degenerate Surface model, so this additional mapping could be enabled by tracking the (u, w) parameters of the points in the intersected surface mesh, and using the same mapping processes that are already applied to the degenerate models. Additional handling would be required for applying analysis results and deflections along the lines of intersection, where the mapping algorithms for the separate components come into conflict.

Acknowledgments

This work was conducted as part of the NASA Transformational Tools and Technologies Project, led by Michael Rogers (acting), within the Multi-Disciplinary Design, Analysis and Optimization element, led by Jeffrey K. Viken.

References

- ¹Raymer, D. P., *Aircraft Design: A Conceptual Approach*, AIAA Education Series, Washington, DC, 1989.
- ²Lazzara, D. S., Haines, R., and Willcox, K., "Multifidelity Geometry and Analysis in Aircraft Conceptual Design," *19th AIAA Computational Fluid Dynamics Conference*, AIAA 2009-3806, San Antonio, TX, 2009.
- ³Hahn, A., "Vehicle Sketch Pad: A Parametric Geometry Modeler for Conceptual Aircraft Design," *48th AIAA Aerospace Sciences Meeting Including the New Horizons Forum and Aerospace Exposition*, AIAA 2010-657, Orlando, FL, 2010.
- ⁴Ordaz, I., "Conversion of Component-Based Point Definition to VSP Model and Higher-Order Meshing," *49th AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition*, AIAA 2011-358, Orlando, FL, 2011.
- ⁵Chaput, A. and Rizo-Patron, S., "Vehicle Sketch Pad Structural Analysis Module Enhancements for Wing Design," *50th AIAA Aerospace Sciences Meeting*, AIAA 2012-0546, Nashville, TN, 2012.
- ⁶Hahn, A., "Application of Cart3D to Complex Propulsion-Airframe Integration with Vehicle Sketch Pad," *50th AIAA Aerospace Sciences Meeting*, AIAA 2012-0547, Nashville, TN, 2012.
- ⁷Belben, J. B. and McDonald, R. A., "Enabling Rapid Conceptual Design Using Geometry-Based Multi-Fidelity Models in VSP," *51st AIAA Aerospace Sciences Meeting Including the New Horizons Forum and Aerospace Exposition*, AIAA 2013-0328, Grapevine, TX, 2013.
- ⁸*ModelCenter, Software Package, Ver. 11.2*, Phoenix Integration, Blacksburg, VA, 2015.
- ⁹*FRICITION, Software Package, Ver. 3*, Virginia Polytechnic Institute and State University, Blacksburg, VA, 2001.
- ¹⁰Drela, M., "XFOIL: An Analysis and Design System for Low Reynolds Number Airfoils," *Conference on Low Reynolds Number Airfoil Aerodynamics*, University of Notre Dame, South Bend, IN, 1989.
- ¹¹Drela, M., "Integrated Simulation Model for Preliminary Aerodynamic, Structural, and Control-Law Design of Aircraft," *40th AIAA Structures, Structural Dynamics, and Materials Conference and Exhibit*, AIAA 991394, St. Louis, MO, 1999.